Amendments to the Specification:

Amend page 2, paragraph beginning at line 22, as follows:

A1

Patent application S.N. 09/548,907 ~~(tba - Docket RAL9-00-0010)~~ filed concurrently by Brian M. Bass et al. and entitled "Method and System for Network Processor Scheduler". This patent is sometimes referred to herein as the Scheduler Structure Patent.

Amend page 2, paragraph beginning at line 26, as follows:

A2

Patent application S.N. 09/548,910 ~~(tba - Docket RAL9-00-0014)~~ filed concurrently by Brian M. Bass et al. and entitled "Method and System for Network Processor Scheduling Outputs Based on Multiple Calendars". This patent is sometimes referred to herein as the Calendar Scheduling Patent.

Amend page 3, paragraph beginning at line 3, as follows:

A3

Patent application S.N. 09/548,911 ~~(tba - Docket RAL9-00-0015)~~ filed concurrently by Brian M. Bass et al. and entitled "Method and System for Network Processor Scheduling Based on Calculation". This patent is sometimes referred to herein as the Calculation Patent.

Amend page 3, paragraph beginning at line 6, as follows:

A4

Patent application S.N. 09/834,141 ~~(tba - Docket RAL9-00-0016)~~ filed concurrently by Brian M. Bass et al. and entitled "Method and System for Network

Processor Scheduling Based on Service Levels". This patent is sometimes referred to herein as the Service Level Patent.

Amend page 3, paragraph beginning at line 10, as follows:

A5 Patent application S.N. 09/548,912 ~~(tba - Docket RAL9-00-0017)~~ filed concurrently by Brian M. Bass et al. and entitled "Method and System for Network Processor Scheduling Outputs Using Queueing". This patent is sometimes referred to herein as the Queueing Patent.

Amend page 3, paragraph beginning at line 15, as follows:

A6 Patent application S.N. 09/546,651 ~~(tba-Docket RAL9-00-0007)~~ filed April 10, 2000 by Brian M. Bass et al. and entitled "Method and System for Minimizing Congestion in a Network". This patent is sometimes referred to herein as the Flow Control Patent.

Amend page 3, paragraph beginning at line 18, as follows:

A7 Patent application S.N. 09/547,280 ~~(tba-Docket RAL9-00-0004)~~ filed April 11, 2000 and entitled "Unified Method and System for Scheduling and Discarding Packets in Computer Networks". This patent is sometimes referred to herein as the Packet Discard Patent.

Amend page 9, paragraph beginning at line 14, as follows:

The present invention overcomes the disadvantages and limitations of the prior art systems by providing a simple, yet effective, way of handling information units or frames coming out of a processing system and directing frames to output ports for dispatch to ~~an~~ a data transmission network. The present invention has particular application to a system in which packets of variable length are being handled from a plurality of users and where a level of service commitment has been made to at least some of the users.

Amend page 10, paragraph beginning at line 3, as follows:

The present invention has the advantage that it allows the efficient use of bandwidth resource and allows service level commitments to be fulfilled while allowing any remaining bandwidth to be used efficiently and equitably. The present invention allows a simple calculation without performing division to support the calculation of a new location in the ~~queue~~ calendar for a serviced flow queue.

Amend page 15, paragraph beginning at line 1, as follows:

The arrows show the general flow of data within the interface system shown in Fig. 1. Frames of data or messages (also sometimes referred to as packets or information units) received from an Ethernet MAC 14 off the ENET PHY block 38 via the DMU bus are placed in internal data store buffers 16a by the EDS-UP device 16. The frames may be identified as either normal frames or guided frames, which then relates to method and location of the subsequent processing in the plurality of processors. After the input units or frames are processed by one of the plurality of processors in the

embedded processor complex, the completed information units are sent to the switch to be delivered to an egress ~~ingress~~ side of a network processor. Once the information units are received on the egress ~~ingress~~ side of the network processor, they are processed by one of the plurality of processors in the embedded processor complex, and when the egress processing is completed, they are scheduled through the scheduler 40 out of the processing unit 10 and onto the data transmission network through the PMM-DN multiplexed MAC's 36 and the physical layer 38.

Amend page 18, paragraph beginning at line 1, as follows:

In each of the above mentioned calendars, a pointer (a Flow ID) is used to represent a flow queue's location within the calendar. Thus, flow 0 has its Flow ID 221 in calendar 220, flow 1 has a FlowID 232 in calendar 230 and FlowID 241 in the WFQ 240 and flow 2047 has a FlowID 231 in calendar 230 and FlowID 251 in calendar 250, all as indicated by the arrows in Fig. 3. Further there may be none, one, or two such pointers to a single flow queue present in the plurality of calendars in the system. Typically, pointers in a calendar ~~to~~ do not represent un-initialized or empty flow queues. When a pointer to a flow queue (or a FlowID) is present in a particular calendar in the system, the flow queue may be referred to as being "in" that particular calendar.

Amend page 18, paragraph beginning at line 17, as follows:

Each of the time-based calendars 220, 230 and 250 consists of a plurality of epochs, with four shown for each in Fig. 3 as represented by the overlapping rectangles. Fig. 4 shows the four epochs 302, 304, 306 and 308 along with a typical timing arrangement for the epochs where the first epoch 302 (labeled epoch0) has a step of the scheduler tick (allowing 512 bytes every 150 nsec in this case), the second epoch

304 has a step of 16 times that of the first epoch 302, with the third epoch 306 having the same ratio to the second epoch 304 and the fourth epoch 308 having the same ratio to the third epoch 306.In this way, the first epoch 302 has a high priority (it is scheduled for service sixteen times as often as the second epoch ~~304), creating~~ 304). This arrangement creates a hierarchy of service priorities which will have associated increases in cost. A current pointer (e.g., 312 for epoch 302) is associated with each epoch to provide a pointer as to where in the queue the processing is currently located. Since the present system of progressing through the epochs is to increment the current pointer, the direction of processing is from lower to higher in the epoch. Also shown in this Fig. 4 is the current time 320 and a scheduler tick 330 which drives the clock 320 as well as driving the steps within each epoch.

Amend page 20, paragraph beginning at line 1, as follows:

When a queue becomes empty, it is removed from the calendars in a form of disconnection. When a queue which is not in any calendar begins to ~~send~~ receive frames, it is treated as a new queue in a process called connection (or re-connection for the return of a queue which has previously been ~~sending frames.)~~ receiving frames). The process of disconnection and then reconnection might have the undesirable result of allowing the queue to be placed at the calendar location pointed to by the current (time) pointer ~~advance to the head of the line~~, in front of (earlier) where it would be if it was continuing in its place with its calendar location ~~priority~~ calculated after each servicing.